ERRATUM

Erratum to: New wireless connection between user and VE using speech processing

M. Ali Mirzaei[1] · Frederic Merienne[1] · James H. Oliver[2] · Jean-Rémy Chardonnet[1]

In the original publication, Dr. Jean-Rémy Chardonnet was not included as a co-author. In recognition of his contribution to the article, the authors wish to add him in the author group. Hence, the corrected author group and his affiliation are as follows:

Corrected author group: M. Ali Mirzaei · Frederic Merienne · James H. Oliver · Jean-Rémy Chardonnet

Co-author Affiliation:
Jean-Rémy Chardonnet,
Arts et Métiers ParisTech, CNRS, Le2i, Institut Image, Chalon-sur-Saône, France,
email: jean-remy.chardonnet@ensam.eu

In addition, the affiliations for M. Ali Mirzaei and Frederic Merienne should be corrected to "Arts et Métiers ParisTech, CNRS, Le2i, Institut Image, Chalon-sur-Saône, France".

✉ M. Ali Mirzaei
mirzai142.nri@gmail.com

James H. Oliver
oliver@iastate.edu

Jean-Rémy Chardonnet
jean-remy.chardonnet@ensam.eu

[1] Arts et Métiers ParisTech, CNRS, Le2i, Institut Image, Chalon-sur-Saône, France

[2] Virtual Reality Application Center, Iowa State University, Ames, IA, USA

New wireless connection between user and VE using speech processing

M. Ali Mirzaei · Frederic Merienne · James H. Oliver

Abstract This paper presents a novel speak-to-VR virtual-reality peripheral network (VRPN) server based on speech processing. The server uses a microphone array as a speech source and streams the results of the process through a Wi-Fi network. The proposed VRPN server provides a handy, portable and wireless human machine interface that can facilitate interaction in a variety interfaces and application domains including HMD- and CAVE-based virtual reality systems, flight and driving simulators and many others. The VRPN server is based on a speech processing software development kits and VRPN library in C++. Speak-to-VR VRPN works well even in the presence of background noise or the voices of other users in the vicinity. The speech processing algorithm is not sensitive to the user's accent because it is trained while it is operating. Speech recognition parameters are trained by hidden Markov model in real time. The advantages and disadvantages of the speak-to-VR server are studied under different configurations. Then, the efficiency and the precision of the speak-to-VR server for a real application are validated via a formal user study with ten participants. Two experimental test setups are implemented on a CAVE system by using either Kinect Xbox or array microphone as input device. Each participant is asked to navigate in a virtual environment and manipulate an object. The experimental data analysis shows promising results and motivates additional research opportunities.

Keywords Speak-to-VR · Wi-Fi network · Speech processing · VRPN server

M. A. Mirzaei (✉) · F. Merienne
Lab. Le2i, Institute Image, Paris-Tech, Paris, France
e-mail: mirzai142.nri@gmail.com

J. H. Oliver
Virtual Reality Application Center, Iowa State University, Ames, IA, USA
e-mail: oliver@iastate.edu

1 Introduction

Since the virtual-reality peripheral network (VRPN) system was released in 2001 (Taylor et al. 2001), many of VRPN servers for different devices have been developed and attached to the library (VRPN 07.30 R. M. T. II 2008). This includes Dream Cheeky USB drum (Boyle 2008), Hillcrest Labs' Free-space (DiVerdi et al. 2006), the Nintendo Wii Remote (Fischbach et al. 2011), 3DConnexion SpaceNavigator, Logitech Magellan, Spaceball 6DOF motion controllers (Stone et al. 2010), Neat Gadget (TNGs) from MindTel (Taylor et al. 2001), fly-stick, mouse (ZHU et al. 2004), keyboard, 3D mouse, the Xbox 360 game controller (Suma et al. 2011) and so on (for more information see R. M. T. II 2008).

The VRPN system provides a device independent and network-transparent interface to virtual reality peripherals and simulation platforms. VRPN now supports thousands of applications in VR research (Taylor et al. 2001) and industrial products such as large-scale simulation, modeling and visualization.

Speech processing technology has progressed recently both in terms of processing speed and accuracy. Different software development kits (SDK) have been proposed by specialists in this arena. A company called iSpeech (2011) has released a free speech processing SDK for mobile developers building apps for iOS, Android and BlackBerry. SAR (2005) proposed an SDK

to enable the engineers to embed speech processing into their products and applications. Intel (2013) has released a SDK for voice processing and synthesis. Microsoft Speech SDK 5.1 (Jinghui et al. 2005) promoted the features of the previous version of the Speech SDK. Now, it is possible to use the Win32, Win64 Speech API (SAPI) to develop speech applications with Visual C++, basic, ECMA-Script and other Automation languages. The SDK also includes freely distributable text-to-speech (TTS) and speech recognition (SR) engines.

Speech-to-VR VRPN server has two distinctive parts, speech processing and streaming through the network. Speech-to-VR incorporates a Wi-Fi network for data streaming since wireless connections are more comfortable for end users. In addition, for CAVE-based applications, the walls of the display make wired connections cumbersome. Thus, speak-to-VR is implemented with Wi-Fi connection to a local area network (LAN).

The aim of the paper was to describe a general purpose and robust speech interface using Microsoft Speech SDK. In particular, speak-to-VR enhances the performance of the Microsoft Speech SDK by implementing a minimum variance distortionless response (MVDR) algorithm for a microphone array. The second objective is to develop a navigation/manipulation application in a VE to test user performance during usage of the speak-to-VR interface. The paper organized as follows: in Sect. 2, the principle of MVDR algorithm is briefly explained and microphone array architecture for speech recognition is introduced. This section also describes how the result of speech processing is converted to VRPN data type and streamed through a wireless network. Section 3 describes the performance of the proposed speech-based HMI will be tested by navigation/ manipulation tasks inside a CAVE-based virtual environment. The result of a subjective user study and some advantages of this speak-to-VR VRPN serve are discussed in Sect. 4. Some useful information about the performance of the speak-to-VR VRPN server is also explained.

2 Theory behind speech recognition based on microphone array

A primary requirement for speak-to-VR is a broad application domain characterized by a less than ideal sound environment that may include noise, echoes and reverberation. Thus, a robust real-time speech processing approach is required that also minimizes delay, in order to facilitate interactive performance. Thus, speak-to-VR is implemented with a microphone array that provides spatially discriminated input to the MVDR algorithm. The channel impulse responses $h_i(r,t)$ describe sound propagation from the source to the individual microphones. The discrete-time beam-former is modeled by fast Fourier transform (FFT) overlap-add filter bank (Pulakka and Alku 2011). The MVDR (Shao and Qian 2013) beam-former algorithm in the frequency domain is used to analyze this multi-channel system. An MVDR beam-former optimizes the power of the output signal under the constraint that signals from the desired direction are maintained (Rubsamen and Gershman 2012). Optimization constraints (Eq. 1) can be solved using Lagrange's method (2):

$$\mathbf{w_o} = \underset{\mathbf{w}}{\text{argmin}} \left\{\mathbf{w^H S_{xx} w}\right\}, \mathbf{with\ w^H h = 1} \tag{1}$$

$$\nabla_w \left[\mathbf{w^H S_{xx} w} + \lambda\left(\mathbf{w^H h - 1}\right)\right] = \mathbf{S_{vv} w} + \lambda \mathbf{h} = \mathbf{0} \tag{2}$$

where $\mathbf{S_{xx}}, \mathbf{w}, \nabla_\mathbf{w}$ are spatio-spectral correlation matrix, beam-former weights and the gradient with respect to the weight vector, respectively. Superscripts H and h denote the conjugate transpose and the channel transfer function vector. Combining the constraint equation from (1) with (2) leads to the well-known solution for the optimum weight vector

$$\mathbf{w}_o = \frac{\mathbf{S}_{vv}^{-1}\mathbf{h}}{\mathbf{h}^H \mathbf{S}_{vv}^{-1}\mathbf{h}} \tag{3}$$

If the noise is assumed as homogeneous diffuse noise and if $\mathbf{S_{vv}}$ is estimated for each signal frame with index m by

$$\begin{aligned}&\mathbf{S_{vv}}\left(\mathbf{e}^{\mathbf{j}\theta}, \mathbf{m}\right)\\ &\quad = \alpha \mathbf{S_{vv}}\left(\mathbf{e}^{\mathbf{j}\theta}, \mathbf{m-1}\right) + (\mathbf{1}-\alpha)\mathbf{v}\left(\mathbf{e}^{\mathbf{j}\theta}, \mathbf{m}\right)\mathbf{v^H}\left(\mathbf{e}^{\mathbf{j}\theta}, \mathbf{m}\right)\end{aligned} \tag{4}$$

where $\theta = 2\pi\frac{f}{f_s}$ is the frequency variable, the optimum weight vector can be found iteratively with a steepest descent algorithm expressed by,

$$\begin{aligned}\mathbf{w_{k+1}} &= \mathbf{w_k} - \mu\nabla_\mathbf{w}\left[\mathbf{w_k^H S_{xx} w_k} + \lambda\left(\mathbf{w_k^H h - 1}\right)\right]\\ &= \mathbf{w_k} - \mu(\mathbf{S_{vv} w_k} + \lambda\mathbf{h})\end{aligned} \tag{5}$$

The Lagrange multiplier, λ, is calculated by substituting the second constraint of (1) into (2). By eliminating λ from (5), the final update equation is given by,

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \mu \underbrace{\left(I - \frac{\mathbf{h}\mathbf{h}^H}{\|\mathbf{h}\|^2}\right)\mathbf{S}_{vv}\mathbf{w}_k}_{(\mathbf{g}_k)} \tag{6}$$

In (6), the weight vector is updated by using $\mathbf{S_{vv}}$ estimated from (4) and iterating in each frame. Furthermore, convergence speed is improved by computing an optimum step size factor μ. Step size is chosen so that it minimizes the noise power at the beam-former output in each iteration.

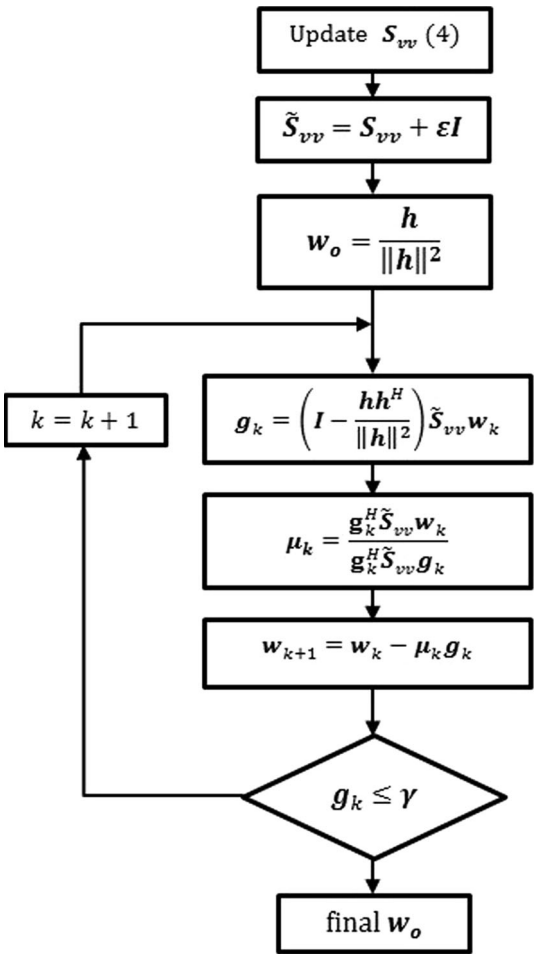

Fig. 1 Flow chart for weight vector calculation

$$\frac{\partial\left(\mathbf{w}_{\mathbf{k+1}}^{\mathbf{H}}\mathbf{S}_{\mathbf{vv}}\mathbf{w}_{\mathbf{k+1}}\right)}{\partial\mu^*} = 0 \tag{7}$$

Combining (6) and (7) yields,

$$\mu_k = \frac{\mathbf{g}_k^H \mathbf{S}_{vv}\mathbf{w}_k}{\mathbf{g}_k^H \mathbf{S}_{vv}\mathbf{g}_k} \tag{8}$$

The entire process is summarized in the flowchart shown in Fig. 1.

3 Implementation of speech processing

The proposed speech recognition system consists of six components as depicted in Fig. 2. Among them, the key signal processing components include feature analysis, unit matching system, lexical coding and syntactic analysis. The speech signal is analyzed in the frequency and time domain to extract observation vectors, which can be used to train a hidden Markov model (HMM) (Lyngsø 2012). The HMM is a useful algorithm for characterizing various speech sounds.

The reason why HMM algorithms are popular in speech processing is because they can be trained automatically and computationally feasible to use. The HMM is an stochastic approach which models the given problem as a "doubly stochastic process". The detail of the HMM training is well described in Nilsson and Ejnarsson (2002).

A speech processing algorithm is a chain of different processes. First, a choice of speech recognition must be made by the unit matching system to detect subwords. Possibilities include linguistically basic subword units as well as derivative units. For the current application, it is both reasonable and practical to consider the word as a basic speech unit. Thus, for the current implementation, only words listed in Table 1 are considered.

The lexical coding process places constraints on the unit matching system so that the paths investigated are those corresponding to sequences of speech units, which are in a word dictionary (lexicon). This procedure implies that the word extracted by speech recognition must be specified in terms of basic units chosen for recognition. Syntactic analysis adds further constraints to the set of recognition search paths. One way in which semantic constraints are utilized is via a dynamic model of the state of the recognizer. For each word recognized, the associated code (fourth row in Table 1) is selected and server data are generated and sent to the client side through the wireless network. Then, the corresponding function is activated in the client application.

4 Experiments and validation setup

Test environment as shown in Fig. 3 was established to evaluate the performance of the proposed VRPN server. The environment consists of CAVE system, a microphone array and its interfacing system, a laptop computer equipped with wireless router, as well as an Ethernet router, LAN Hub and VE workstation computer along with its accessories. A test software platform was developed to manage the CAVE display system. The platform uses OpenSceneGraph to render the 3D model and incorporate the properties of the virtual environment into the model. The model is projected in the display system via MPI and four NVidia Quadroplex GPUs. All the C++ functions were wrapped under Java script functions in the platform to make development faster and easier for programmers.

Every microphone array can be used as audio input device; however, Kinect Xbox 360 is used in these experiments. The microphone array features four microphone capsules (Joystiq 2011; Store 2010) and operates with each channel processing 16-bit audio at a sampling

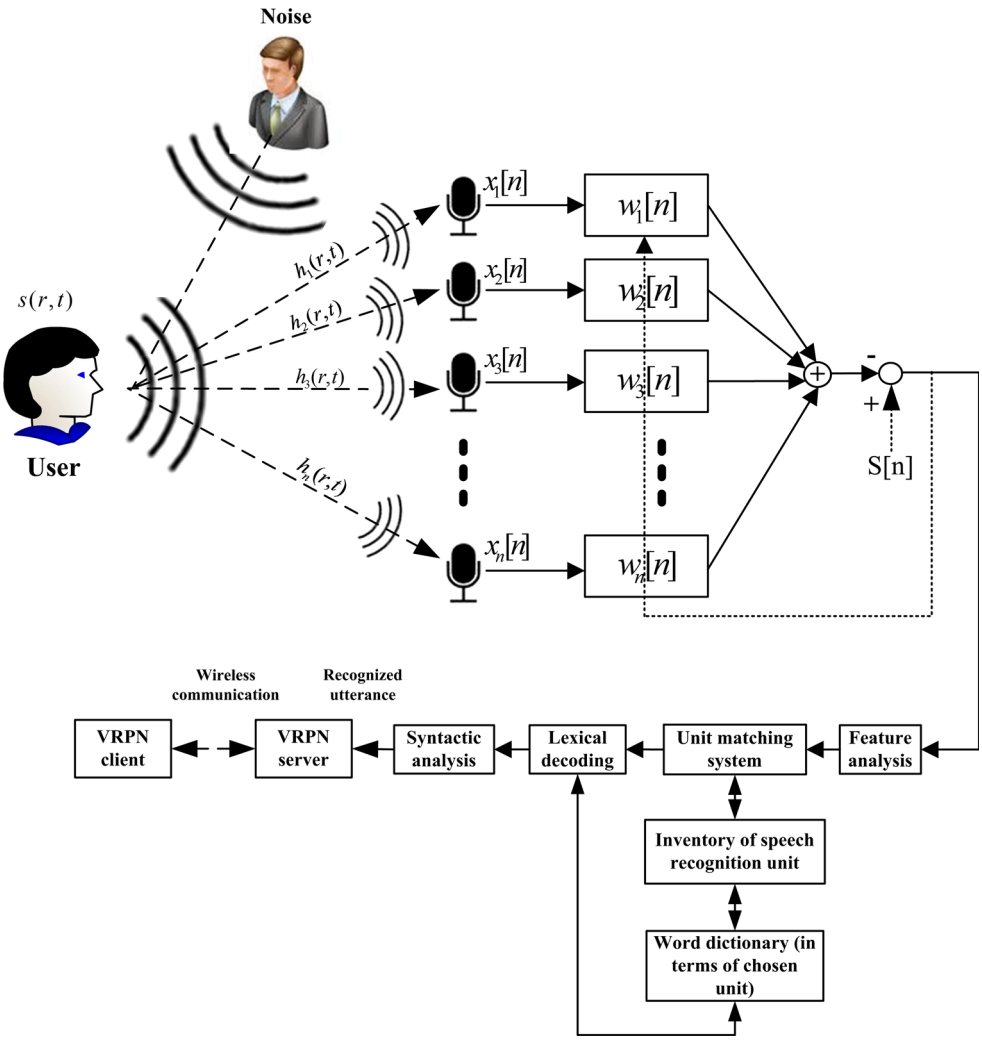

Fig. 2 The architecture of speech recognition and coding system

Table 1 The vocabulary list in the dictionary of speech processing system (lexicon)

Task	Vocabulary						
Navigate	Begin	Start	Stop	Forward	Backward	Turn right	Turn left
Interact	–	–	–	select	rotate	move	place
Code	00000	00001	00010	000100	001000	010000	100000

rate of 16 kHz (Retrieved 2010). The distance between the user and the microphone array is limited to 1–3 meters because the array is located inside the CAVE. The speak-to-VR server ran on a Dell Precision laptop (Intel Core^{TM} i7 processors, RAM 32GB 1333MHz, 7200rpm hard drive 350GB, graphics NVIDIA Quadro 1000M, integrated noise canceling digital array microphones, USB 2.0 and IEEE 1394 I/O ports, Windows 7 Professional 64-Bit) and streamed the Speech Processing (SP) result as 7 buttons over a Wi-Fi network. VRPN has different data type such as "Button", "Digital", "Tracker" and so on. Button data type is selected because it is easier for the development. The microphone array and the distance between the elements of the array are shown in Fig. 4.

The speak-to-VR VRPN server processes spoken commands and compares the result with the dictionary content (Table 1) after receiving a word as a 1D signal from microphone array. Then, the associated button is activated and sent to the client application through the wireless network. Navigation and object manipulation applications were developed in the client side to evaluate user performance. Since it is difficult to use only the speak-to-VR interface for both navigation and object manipulation, this interface was used in combination with other HMI. The

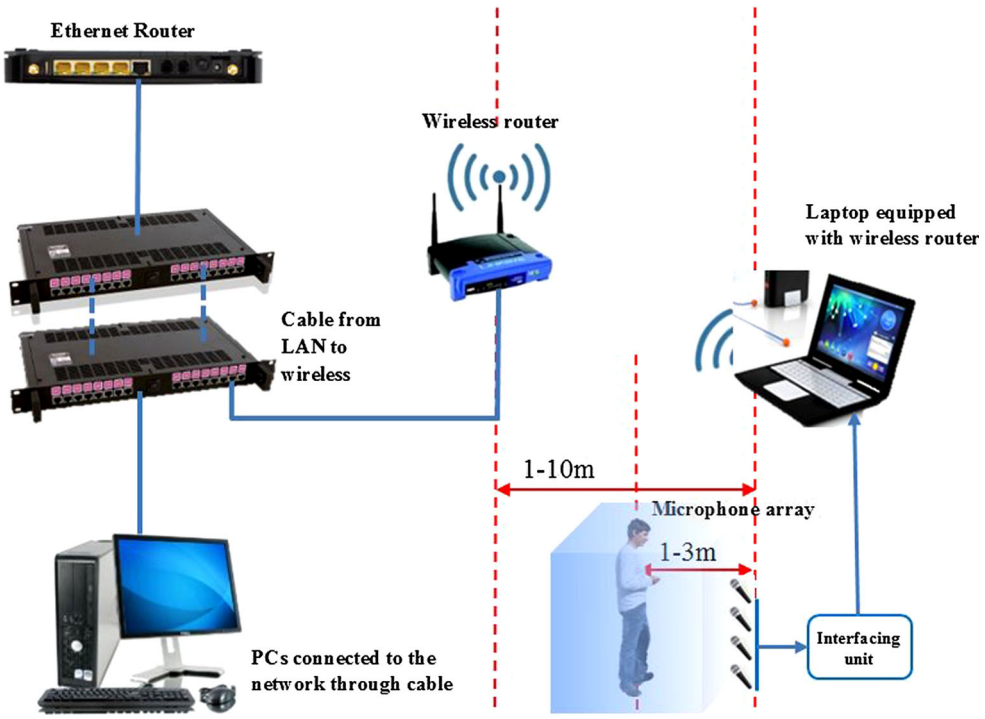

Fig. 3 Test-bench and wireless infra-structure for voice data streaming in speak-to-VR VRPN server

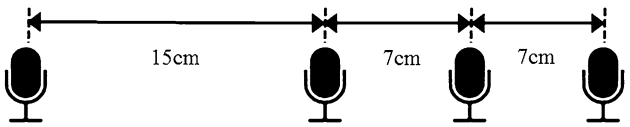

Fig. 4 Micro-phone array configuration

detail of these two tasks will be explained separately below.

4.1 Navigation

Navigation in large-scale 3D VEs can be implemented by the combination of two types of movement, i.e., translation and rotation. These movements can be initiated either by clicking on a button or pressing a joy-stick handle when a navigation device such as fly-stick is implemented. Similarly, the movement can be initiated and terminated by using speak-to-VR VRPN server buttons (voice commands) as explained above (Table 1). We asked the user to go from position A to B using two interfaces (Fig. 5). In the first interface, the task is performed by using only "start", "stop", "turn left" and "turn right" commands from speak-to-VR VRPN server. In the second interface, "start" and "stop" commands from speak-to-VR and rotation from AR-Tracker (ZHU et al. 2004) VRPN server were employed to complete the navigation task. The screen shot of the server and client side of speak-to-VR VRPN are shown at the top right corner in Fig. 5. As seen, the server side has sent a code associated with "backward", in "button 3", by setting 4th digital bit active. Consecutively, "Button 1" became active when "stop" had been told. In the client side, the assigned function starts navigation/manipulation task immediately after receiving the code from speak-VRPN server.

4.2 Object manipulation

Object manipulation was implemented by fusing data from speak-to-VR and a Magic bracelet VRPN server. The Magic bracelet is shown in Fig. 6. For the manipulation task, the user is asked to "select", "move" and "place" an object (a cube of 30 × 30 × 30) from location A to B. The object is selected by saying "select" when the bracelet is close enough (30 cm) to the target object. Then, the object and the bracelet become one object and by the movement of the bracelet it moves. The object is placed in the last location of the bracelet when the command "place" is said by the user.

5 Result and discussion

The dictionary is not limited to the vocabulary listed in Table 1. It can be extended, but a larger dictionary and longer words increase the processing time significantly. Tables 3 and 4 show the relation between speech processing time (SPT) and total processing time (TPT) with

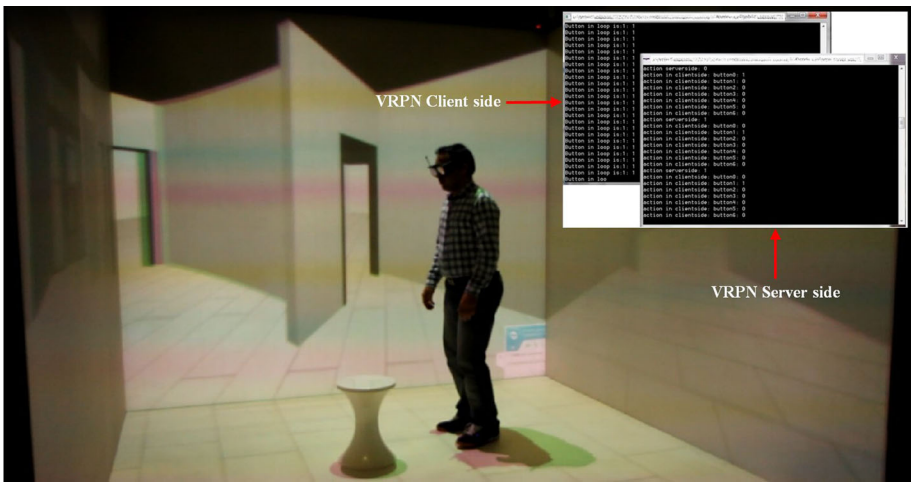

Fig. 5 Navigation inside a real-scale 3D model using speak-to-VR as an input device

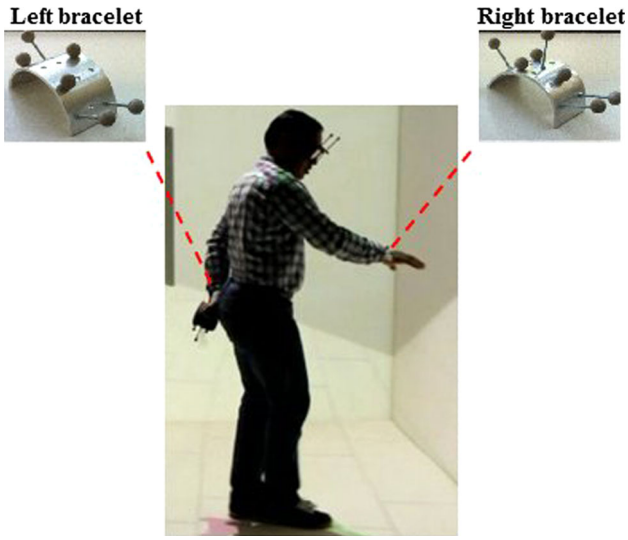

Fig. 6 Pair of magic bracelets

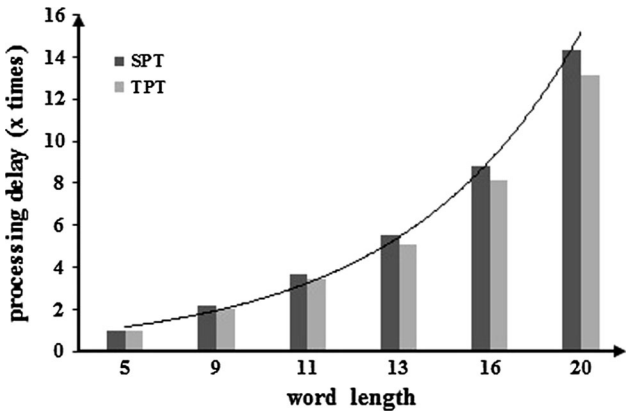

Fig. 7 Relation between the length of the words and reduction in the processing speed (increasing processing delay)

the length and the number of the words in the dictionary, respectively. The SPT is only the time speech takes to be processed, and the TPT includes the speech processing time plus the transit time in the VRPN network. To compute these times, one high-resolution timer is started when the speech signal received from the microphone array and stopped when the 7-bit code is received by to the client application. The difference between these two instances is total processing time. The same strategy was used to measure the speech processing time.

5.1 Parameters study

Two dictionaries have been chosen to study the usability and the restriction of the proposed speech-based HMI for the navigation task, dictionary 1 (15 words with min 5 and max 20 letters) and dictionary 2 (8 words with min and max similar to dictionary 1) as seen in Table 2.

Table 2 Two dictionaries used in the navigation study

Task	Vocabularies of navigation dictionary
Dictionary 1	Start, stop, turn left, turn right, forward, backward, move forward, move backward, increase speed, decrease speed, decrease velocity, increase velocity, increase acceleration, decrease acceleration
Dictionary 2	Start, stop, turn left, turn right, move forward, decrease velocity, increase speed, increase acceleration

As expected, speech recognition performance decreases with larger dictionary. For example, for dictionary 1 SPT and TPT for "start" in dictionary 1 are 106 (ms) and 177(ms), respectively, while for dictionary 2 they are 82(ms) and 141(ms). The SPT increases 29.3 % when the number of the words in the dictionary is doubled (Table 3 and 4). The difference of two processing time is calculated

Table 3 Speech processing and total processing time for navigation dictionary 1 in Table 2

Length of words	5	9	11	13	16	20
Sample word	Start	Turn right	Move forward	Increase speed	Decrease velocity	Increase acceleration
Speech processing time (ms)	106	334.8	497.2	691.6	1,043.2	1,624
Total processing time (ms)	177	534.6	785.4	1,084.2	1,622.4	2,508

Table 4 Speech processing and total processing time for navigation dictionary 2 in table 2

Length of words	5	9	11	13	16	20
Sample word	Start	Turn right	Move forward	Increase speed	Decrease velocity	Increase acceleration
Speech processing time (ms)	82	248.4	365.2	504.4	755.2	1,168
Total Processing time (ms)	141	405	587.4	803.4	1,190.4	1,824

Table 5 Two dictionary used in manipulation study

Task	Vocabulary in the manipulation dictionary
Dictionary 1	Select, move, rotate up, rotate down, rotate clockwise, rotate counterclockwise, change color, change shape, change spawned object
Dictionary 2	Select, move, rotate up, rotate down, rotate clockwise, rotate counterclockwise

Table 6 Speech processing and total processing time for manipulation dictionary 1 in table 5

Length of words	4	5	6	8	10	16	20
Sample word	Move	Place	Select	Rotate up	Rotate down	Rotate clockwise	Rotate counterclockwise
Speech processing time	66	101.25	211.2	252	390	984	1,530
Total Processing time	109.2	162.75	243	386.4	588	1,444.8	2,226

by $(SPT_2 - SPT_1)/SPT_1 \times 100\,\%$ and $(TPT_2 - TPT_1)/TPT_1 \times 100\,\%$ where subscripts 1 and 2 indicate the dictionary. TPT increases 25.5 %, for the same word, by increasing the number of the words. On average, SPT and TPT increase 35.7 and 33.3 %, respectively, when the number of the words is doubled. However, when the length of the words (#letters) increases in the same dictionary (e.g. dictionary 1), SPT and TPT increase quite rapidly.

SPT associated with "start" and "Increase speed" (with 5 and 13 letters) is 106(ms) and 691.6(ms), respectively, which leads to 5.5 times slower processing. TPT becomes 5.1 times slower for the same words. As seen, TPT and SPT get 5 times slower when the length increases almost 2.5 times. Figure 7 shows the summary of this reduction in TPT and SPT for all the words in Table 3. All of the words in Table 3 were compared to the word "start" to calculate values in Fig. 7. Clearly, the relation between the length of the words and TPT and SPT increase is approximately exponential which in turn means the length of the word is extremely important and the words need to be carefully selected.

Similarly, two dictionaries have been chosen for manipulation task, dictionary 1 (9 words with min 4 and max 20 letters) and dictionary 2 (6 words with min and max similar to dictionary 1) as seen in Table 5. STP and TPT for "place" (with 5 letters in length) increase 26.9 and 21.9 % when vocabulary increases by 1.5 times (Tables 6 and 7). Comparison of "move" and "Rotate clockwise" shows SPT and TPT increase 13.9 and 12.2 times, respectively. Similar to the navigation dictionary, the processing time grows exponentially with the length of words. Since cyber sickness is frequently attributed to lag time between input and response (Day et al. 2001), appropriate selection of vocabulary for navigation and manipulation is critical.

Based on this heuristic, more than 10 words in the dictionary make the server quite slow. Moreover, the length of the words is important, words longer than 10 letters decrease the processing speed dramatically. As a

Table 7 Speech processing and total processing time for manipulation dictionary 2 in Table 5

Length of words	4	5	6	8	10	16	20
Sample word	Move	Place	Select	Rotate rip	Rotate down	Rotate clockwise	Rotate counterclockwise
Speech processing time	52	77.5	108	184	280	688	1,060
Total processing time	89.6	129.5	176.4	291.2	434	1,030.4	1,568

Table 8 Random accuracy test of speak-to-VR VRPN server

Word 1 Word 2	Start Stop	Left Right	Increase Decrease	Forward Backward
Number of repetition	40	30	20	40
Precision 1	95.6	98.2	93.2	97.6
Precision 2	94.8	97.7	94.5	98.4

result, it is recommended to select shorter words and include fewer words in the dictionary to make it as fast as possible.

The precision of the server is also important. The precision is defined as the number of successful instances of speech recognition compared to the total number of trials. To test the precision of the speak-to-VR VRPN server, four pairs of words were selected and spoken in front of the array while the detection result was recording. The test word pairs were chosen to be intentionally close in terms of pronunciation. The result of this test is shown in Table 8. The recognition rate is high on average ($M = 96.25$ %); however, when the words are very close, for instance "start" and "stop", the detection rate decreases somewhat (95.6, 94.8 %). On the contrary, the recognition precision increases when the words are different, for instance "left" and "right" (98.2, 97.7 %).

5.2 Evaluation by subjective user study

A subjective user study was conducted to evaluate the user performance of the speak-to-VR VRPN server. A simulator sickness questionnaire (SSQ Kennedy et al. 1993) and postexposure question were used as two evaluation methods. Ten subjects (6 males and 4 females: ages 31.58 ± 12.69 years and weights 74.65 ± 15.22 kg) participated in the experiment. They performed two tasks as explained above: navigation and manipulation. Participants were asked to score navigation/manipulation tasks using speak-to-VR HMI on a Likert scale of 1-10 (1: bad, 10: very good) in the postexposure questionnaire. Figure 8 shows the result of this evaluation.

A dependent-samples t test was conducted to compare manipulation ($M = 8.4$, SD = 1.43) and navigation ($M = 6.5$, SD = 1.96). This analysis yields $t(9) = 2.3$, $P = 0.046$ and since $p \prec 0.05$ the difference is significant. As seen in Fig. 8, most of the participants found manipulation more comfortable with the speak-to-VR interface than navigation. This is because, there are substantial delays using only the speak-to-VR HMI which induces general discomfort for users when they are navigating. To avoid this general discomfort, we combined the interface with AR-Tracker as mentioned above.

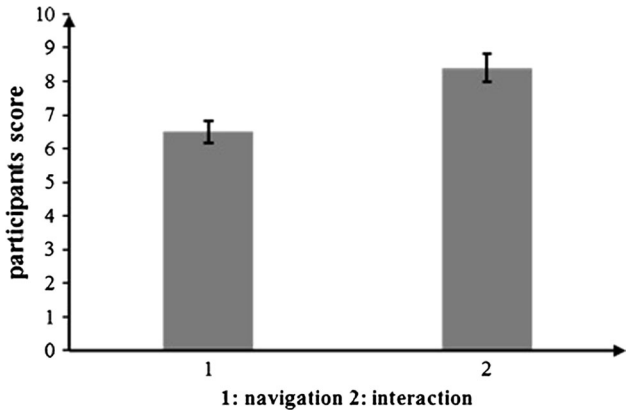

Fig. 8 The level of user satisfaction using speak-to-VR interface for interaction and navigation

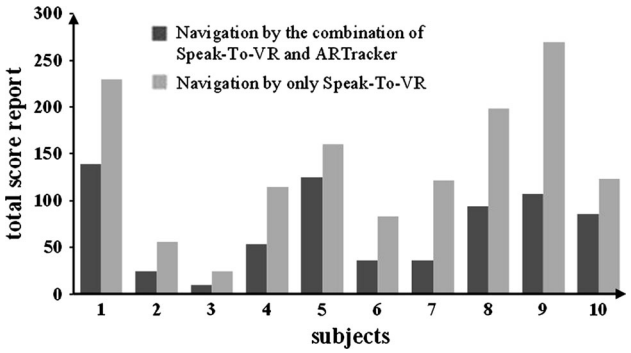

Fig. 9 Comparison of two navigation interface based on speak-to-VR VRPN server

The postexposure sickness score after navigating by the first and the second interfaces (see Sect. 4.1) for each participant is illustrated in Fig. 9. A dependent-samples t-test was conducted to compare navigation using only the speak-to-VR interface (M = 138.2, SD = 72.2) and navigation by combining two interfaces ($M = 71.2$, SD = 44.9),

speak-to-VR and AR-Tracker, via Kennedy SSQ (Kennedy et al. 1993). The analysis yields $t(9) = 4.76$, $P = 0.0098$. this means, in average, participants felt less discomfort with the combination of two HMI.

In addition to the previous evaluations, the accuracy of the speech-based HMI is evaluated in the presence of the accent changes. Four subjects with different accents (Italian, Persian, Indian, Chinese) were selected and were trained for nearly 10 min. They were asked to repeat words "start", "stop", "turn left" and "turn right" several trial (35 times) during training phase and real test. As expected, the precision of the speak-to-VR for training and real test is above 85 and 94 %, respectively. Moreover, the difference between subjects is only ($Min = 94.3\,\%, Max = 97.14\,\%$) 3 %, which means when the accent changes the precision does not change significantly.

6 Conclusion

The speak-to-VR VRPN server described in this paper is based on the Microsoft speech processing SDK and uses a Wi-Fi network to provide a handy, portable and wireless connection between user and a VR system. The subjective user study shows that this HMI is more useful for interaction tasks ($t(9) = 2.3$, $p \prec 0.05$). However, if this HMI (for translation) is combined with another HMI (rotation), it can be used for navigation as well ($t(9) = 4.76$, $p \prec 0.001$). The proposed VRPN server works independent of user's accent and functions precisely (higher than 90 % on average) with the presence of other users and background noise.

7 Future research

The delay of the speak-to-VR VRPN server restricts the usage of the speech-based HMI in a real-time VR applications. Rodriguez et al. (2001) proposed a reconfigurable FPGA-based architecture to accelerate speech recognition on a FPGA. Therefore, one solution for this problem is to implement speech processing on a hardware such as a GPU or FPGA with parallel processing approaches. Then, this solution can be combined with VRPN server to make the speak-to-VR faster for a real-time application.

References

Boyle A et al (2008) Pick your top geek gift-cosmic log. Science 3147:10

Day PN, Holt PO, Russell GT (2001) The cognitive effects of delayed visual feedback: working memory disruption while driving in virtual environments. Cognitive technology: instruments of mind. Springer, Berlin, pp 75–82

DiVerdi S, Rakkolainen I, Höllerer T, Olwal A (2006) A novel walk-through 3d display. In: Electronic imaging 2006, p 605519. International Society for Optics and Photonics

Fischbach M, Wiebusch D, Giebler-Schubert A, Latoschik ME, Rehfeld S, Tramberend H (2011) Sixton's curse-simulator x demonstration. In: 2011 IEEE virtual reality conference (VR), pp 255–256. IEEE

Intel. Voice recognition and synthesis using the intel perceptual computing sdk, 2013

iSpeech. Speech processing sdk for mobile developer, 8 2011

Jinghui G, Zijing J, Jinming H (2005) Implement of speech application program based on speech sdk [j]. J Guangxi Acad Sci 3:169–172

Joystiq. Kinect: the company behind the tech explains how it works, March 21 2011

Kennedy RS, Lane NE, Berbaum KS, Lilienthal MG (1993) Simulator sickness questionnaire: an enhanced method for quantifying simulator sickness. Int J Aviat Psychol 3(3):203–220

Lyngsø R (2012) Hidden Markov models. Narotama 1:1–24

Nilsson M, Ejnarsson M (2002) Speech recognition using hidden Markov model. Master's thesis, Department of Telecommunications and Speech Processing, Blekinge Institute of Technology

Pulakka H, Alku P (2011) Bandwidth extension of telephone speech using a neural network and a filter bank implementation for highband mel spectrum. IEEE Trans Audio Speech Lang Process 19(7):2170–2183

R. M. T. II. (2008) Vrpn 07.30—http://www.cs.unc.edu/research/vrpn/

Retrieved P (2010) Kinect xbox 360 specification, July 2 2010. This information is based on specifications supplied by manufacturers and should be used for guidance only

Rodríguez-Andina J, Fagundes RDR, Junior DB (2001) A fpga-based viterbi algorithm implementation for speech recognition systems. In: 2001 IEEE international conference on acoustics, speech, and signal processing, 2001 (Proceedings ICASSP'01), vol 2, pp 1217–1220, IEEE

Rubsamen M, Gershman AB (2012) Robust adaptive beamforming using multidimensional covariance fitting. IEEE Trans Signal Process 60(2):740–753

SAR (2005) Sri language modeling toolkit and speech sdk

Shao W, Qian Z (2013) A new partially adaptive minimum variance distortionless response beamformer with constrained stability least mean squares algorithm. Adv Sci Lett 19(4):1071–1074

Stone JE, Kohlmeyer A, Vandivort KL, Schulten K (2010) Immersive molecular visualization and interactive modeling with commodity hardware. In: Advances in visual computing. Springer, Berlin, pp 382–393

Store M (2010) Kinect for xbox 360, 7 July 2010. Array of 4 microphones supporting single speaker voice recognition

Suma EA, Lange B, Rizzo A, Krum DM, Bolas M (2011) Faast: The flexible action and articulated skeleton toolkit. In: 2011 IEEE virtual reality conference (VR), pp 247–248, IEEE

Taylor II RM, Hudson TC, Seeger A, Weber H, Juliano J, Helser AT (2001) Vrpn: a device-independent, network-transparent vr peripheral system. In: Proceedings of the ACM symposium on Virtual reality software and technology, pp 55–61, ACM

Zhu F-W, Li D-Q, Yuan Z-P, Wu J-Q, Cheng X (2004) An ar tracker based on planar marker. J Shanghai Univ (Nat Sci Ed) 5:005